

[illegible]

Spring Boot

- Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run"
- Create stand-alone Spring applications
- Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
- Automatically configure Spring and 3rd party libraries whenever possible
- Absolutely no code generation and no requirement for XML configuration

Spring Boot



Spring Boot is not



Not an app or
a web server

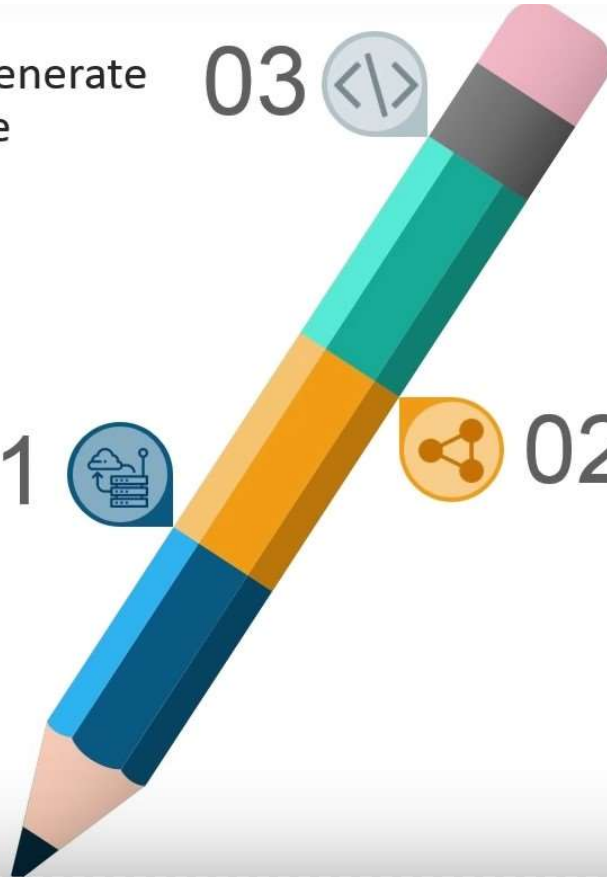
Does not generate
code

03 

01 

02 

Does not implement
any specific
framework



Spring vs Spring Boot

Spring

03

Takes time to have Spring application up and running

02

Manages life cycle of Java

01

Dependency Injection Framework

Spring Boot

03

Shortest way to run Spring application

02

Need not worry about configuring a data source

01

Pre-configured set of frameworks/technologies

Spring Boot Advantages

- Create stand-alone Spring applications that can be started using `java -jar`.
- Embed Tomcat, Jetty or Undertow directly. You don't need to deploy WAR files.
- It provides opinionated 'starter' POMs to simplify your Maven configuration.
- It automatically configure Spring whenever possible.
- It provides production-ready features such as metrics, health checks and externalized configuration.
- Absolutely no code generation and no requirement for XML configuration.

Spring Boot Web

- Provides embedded tomcat server
- Web application can be bundled as normal application with main method
- No DispatcherServlet configuration required
- Configuration can be provided through properties

application.properties

- Used to provide configuration
- Full list of properties at --
- <https://docs.spring.io/spring-boot/docs/current/reference/html/common-application-properties.html>
- Example
 - environments.dev.url=https://dev.example.com
 - environments.dev.name=Developer Setup
 - environments.prod.url=https://another.example.com
 - environments.prod.name=My Cool App

application.yml

- YAML is a superset of JSON, and as such is a very convenient format for specifying hierarchical configuration data
- Example

```
environments:
  dev:
    url: https://dev.example.com
    name: Developer Setup
  prod:
    url: https://another.example.com
    name: My Cool App
```

properties OR yml

- We can use either properties file or yml file.
- Try not to use both in the same application
- If both properties and yml files are available, property defined in properties file will have precedence

Spring Boot JPA

- Automatic CRUD Repository for entities
- CrudRepository interface provides basic methods
- JpaRepository interface provides additional JPA related methods
- DB configuration through properties file or YML file
- In memory database H2

H2 in-memory database

- H2 is in-memory database
- Need H2 dependency
- H2 console can be used by setting following properties

`spring.h2.console.enabled=true`

`spring.datasource.url=jdbc:h2:mem:testdb`

`spring.data.jpa.repositories.bootstrap-mode=default`

CrudRepository

long	count()	Returns the number of entities available.
void	delete(T entity)	Deletes a given entity.
void	deleteAll(Iterable<? extends <u>I</u> > entities)	Deletes the given entities.
void	deleteById(ID id)	Deletes the entity with the given id.
boolean	existsById(ID id)	Returns whether an entity with the given id exists.
Iterable<T>	findAll()	Returns all instances of the type.
Iterable<T>	findAllById(Iterable<ID> ids)	Returns all instances of the type T with the given IDs.
	More methods

Spring Boot JPA properties

```
spring.datasource.url= jdbc:mysql://localhost/ramana
spring.datasource.username=root
spring.datasource.password= rasaspsi
spring.datasource.driver-class-name= com.mysql.jdbc.Driver

spring.jpa.hibernate.ddl-auto= create-drop
```

CrudRepository - Example

Extend CrudRepository and add additional methods if needed

@Repository

```
public interface StudentRepository extends CrudRepository<Student,  
Long>
```

```
{  
    public List<Student> findById(long id);
```

```
    @Query("select s from Student s where s.age <= ?1")  
    public List<Student> findByAgeLessThanEqual (long age);  
}
```

Using CrudRepository

CrudRepository can be autowired in a service or DAO class

`@Autowired`

```
private StudentRepository studentRepository;
```


JPA config annotations

- If the entities are not in the base package, we may have to provide `@EntityScan` annotation to scan for entities

`@EntityScan(basePackages = "com.boot.examples.entity")`

- If the Repositories are not in the base package, we may have to provide `@EnableJpaRepositories` annotation to scan for Repositories

`@EnableJpaRepositories(basePackages = "com.boot.jpa.repo")`

Creating Spring Boot Application

- Through Spring Initializr
- As a maven project
- Using Gradle
- As a starter project in STS